

Development of a Generic XML Personality Metadata Handler for Distributed Entertainment Services

Simon Reymann, Jakub Rachwalski, Stefan Kemper, and Artur Lugmayr

NAMU Lab., Tampere University of Technology
<http://namu.cs.tut.fi/>
FIN-33101 Tampere, +358 (0) 50 935 6668
{simon.reymann, stefan.kemper}@tut.fi,
jakub.rachwalski@gmail.com,
lartur@acm.org

Abstract. Especially in the case of entertainment services, the provision of a general valid personality profile is gaining in importance. The personality profile shall include information about the usage-context of media services. These media services can be arbitrary and range from simple playlist compilations of audio-visual content to complex feedback profiles in interactive television. This research paper provides an overview of a metadata based solution for merging personalized content. Multiple applications, such as audio or video players contribute with user-context information enabling future personalization scenarios. An XML metadata specification is presented providing a generic container format for representing a personality user profile. To enable generality, the personality profile simply extends existing XML standards by adding attributes and thus keeping the existing structure intact. This also allows the creation of slim and resource efficient personality schemas which are especially beneficial when deployed on resource-sensitive client devices like mobile phones.

Keywords: Profiling, Personalization, Personal Content, Metadata Handling.

1 Introduction

Nowadays people are surrounded by multiple electronic devices which, with an increasing probability, hold a personality based profile of a user. This particular piece of information could be used by various applications creating new and customized services. Due to the variety of sources, incompatibilities need to be resolved and a smooth combination needs to be created.

Profiling has been in development for years and is considered as the antidote for the rapid growth of information overrun of e-services. Profiling technologies have been widely described, e.g. in [1] or [2] and also profile merging algorithms from multiple sources have been discussed [3]. However, problems in gathering information from multiple sources and producing personal content [10] of unknown structures are not solved yet. Additional and comparable works are dealing with very application specific solutions as e.g. personalization in the context of broadcasting, applications of specific standards such as MPEG-7 for personalization, specific personalization algorithms, or agent based approaches for personalization. This work is based on the

architectural approach proposed by the Portable Personality (P2) project [4] embedded in the New Ambient MUltimedia Lab. at the Tampere University of Technology [5]. The key-feature of P2 is its general applicability in many application scenarios, especially on the audio-visual sector. The goal of the project is to bridge multiple personal content providers with contemprable consumers evolving into a distributed profiling network with many nodes. In the P2 context, nodes are every-day portable devices like laptops and mobile phones, or stationary systems like desktop PCs and terminals. The following presented solution describes how P2 manages personal information collected from multiple contributing sources and what kind of extension mechanisms are needed to be able to identify and merge personal content from multiple providers.

2 System Architecture

The P2 network consists of three types of nodes, namely providers, consumers and daemons. The structure of the system together with the flow of the information is presented in Fig. 1.

Due to its features and popularity the XML format is the only accepted format in P2. The elements of the system which are responsible for data acquiring are called providers. They send personal content together with P2 management functions to a specific daemon. The daemons store the data, apply the functions provided and send the acquired information to contemprable consumers.



Fig. 1. Overview of the P2 system architecture with multiple provider and consumer nodes interconnected through the distributive P2 daemon network

The P2 architecture could be easily applied on a home entertainment environment, consisting of a digital TV, a Bluetooth enabled set-top-box and multiple Bluetooth enabled mobile phones representing the audience. The mobile devices carry information about the general audio-visual behaviour of its user, such as genre, theme or consumption. That means that each phone represents a P2 daemon with a P2 provider. The set-top-box instead detects all mobile phones within its vicinity and plays a central role in this scenario. On the one hand it behaves like a P2 daemon and collects

available data from surrounding mobile phones via Bluetooth, i.e. it gathers audio-visual behaviour patterns from the audience and merges each individual profile to one representative audience profile. On the other hand the set-top-box is a P2 hybrid being provider and consumer at the same time. Based on the generated audience profile an integrated recommender system could provide a personalized selection of available audio-visual content (P2 consumer). Then, based on the performed decisions of the audience the existing profile can be updated (P2 provider). Additionally, individual profile updates are distributed to each mobile phone of the audience. By making personal content identifiable and mergable a set-top-box is able to pro-actively establish a personalized entertainment environment.

Every XML package received by a daemon may be very distant in structure, as it could come from different providers. What is more, each contained personal content item needs to be identified before it is processed by the daemon according to given management rules. In order to process an unknown standard the personal content provider needs to identify a personal item by extending it with specific P2 attributes (from the `xmlns:p2="http://www.portable-personality.org/"`) before sending it to the daemon. If the daemon knows a standard it is able to identify items and extend them automatically by default rules. As the provider or the daemon knows the structure and extensions he can easily create a complaint version of the XML schema accordingly. Furthermore, when sending personal content to a consumer the daemon deletes all P2 attributes generating valid XML content. Known standards, like Friend of a Friend [6], Description of a Career [7] or Media RSS [8] could be processed automatically.

This is an example of XML content with a specific P2 attribute extension.

```
<favorites p2:item="1">
  <show p2:item="1" p2:limit="title,1,9">
    <title p2:merge="title,1">Found</title>
    <director
p2:merge="director,1">Spielberg</director>
    <value p2:merge="value,3">2.5</value>
  </show>
  <show p2:item="1" p2:limit="title,1,9">
    <title p2:merge="title,1">Spiderwoman</title>
    <director
p2:merge="director,1">Spielberg</director>
    <value p2:merge="value,3">3.3</value>
  </show>
</favorites>
```

Each required management function has its own P2 attribute defined. The attributes' values consist of coma separated fields (CSF) defining options for a single personal content item of the original XML content. If the attribute refers to a couple of elements each field is separated by a semicolon. Following attributes types are valid:

- Single (S SYNTAX), e.g. `p2:item-` with 1 VALUE FIELD or V FIELD;
- Comma separated pairs (CSP SYNTAX), e.g. `p2:merge-` with 2 fields in CSF;
- Comma separated thirds (CST SYNTAX), e.g. `p2:limit-` with 3 fields in CSF;
- Comma separated quads (CSQ SYNTAX), e.g. `p2:delete-` with 4 fields in CSF.

The first item in a CSF is called REFERENCE (R) FIELD, the second MODE (M) FIELD, the third ADDITIONAL (A) FIELD and the fourth EXTRA (E) FIELD. A P2 attribute placed in an element with an R FIELD may refer to the element, its value or a child element. However, if it is an attribute it contains the name of the attribute preceded by \$, e.g. \$<attribute_name>. What is more, the default value of P2 attributes may be changed for the element and all its values, attributes and the child nodes. In this case the element is preceded by @, e.g. @<element_name>.

3 Personality Profile Handling

This section describes the changes of the metadata structure and possible operations on the personality profile, especially merging.

In XML content some of the elements together with their child nodes should be treated as one piece of information (further referred as ITEM). In the exemplary code each “show” element represents an ITEM and is then described by several child nodes like “director” or “title”. The element must be marked with the p2:item="1" attribute which is of S SYNTAX (using p2:item="0" ignores the element and is considered as default). ITEMS represent an actual personal content item and hold all required rules for handling contained data. In some XML structures, like iTunes XML [9] in which the parent-child relationship is not defined by its structure the usage of p2:key, for parent nodes, and p2:parentkey, for child nodes was invented. With this, parent-child relationships are uniquely identified.

3.1 Merging

The profile merging is one of the most important features in the P2 system. In order to merge two ITEMS the system needs to know what ITEMS are of the same nature, i.e. what ITEMS representing the very same personal content but holding different values. It is important to identify the type of an ITEM before merging it with an existing ITEM. Due to the fact that two ITEMS can be of a different XML structure, an identification and merging attribute is implemented. The identification of an ITEM merge key (identification key) is done by using the p2:merge="<REFERENCE>,1" attribute which defines <REFERENCE> as an identifier of the ITEM. All values of all defined identifiers and their location within the structure must be equal before declaring two ITEMS to be of the same type. After characterization through the identifiers the ITEM's child elements get merged according to specified merging modes, each of them representing an individual merging strategy. These in fact can be of any type, e.g. sum, multiply, take the newer, concatenate etc.

For merging, a daemon must hold information about the provider source and details about its type. The source is provided as a Globally Unique Identifier (GUID) and the type is represented by its namespace. Content of multiple sources (multiple GUIDs) embedded in the same namespace specification is merged into one XML representation. Merging of personal content originating from two different namespaces, instead, is not supported. The final result is a collection of multiple namespaces holding personal content of multiple profiling sources.

3.2 Sorting, Limiting and Deleting

Due to the fact that the storage capacity of a daemon is finite, e.g. on mobiles, the provider may set rules when to remove an ITEM using additional P2 attributes. The first option could be deleting on limit in which the number of ITEMS contained in the profile could be limited up to a certain value using the p2:limit attribute (CST SYNTAX). The limitation takes effect on every ITEM which lies at the same location within the XML structure holding the same ITEM element name and having the same p2:limit attribute defined. The M FIELD defines the sorting mode, e.g. ascending or descending. The R FIELD refers to the element on which the sorting is based on while the A FIELD defines the limitation value, e.g. 100. A second option is deleting on a value by using the p2:delete attribute (CSQ SYNTAX). An ITEM may be deleted when the value of a certain attribute or child element reaches a limit. The M FIELD holds the condition to be met, e.g. =, <, >, <= or >=, the A FIELD states the limit value and the E FIELD provides grouping functionalities meaning that the conditions of all group members need to be met before deleting. With this, logical AND and OR operations are implemented.

4 Conclusions

The proposed solution for handling personal content based on different XML standards is straight forward and resource-effective introducing small overhead. Based on simple attributes, XML content can be easily extended enabling features like personal content identification and also merging, sorting, limiting and deleting.

As being part of the P2 public domain forum “*www.portable-personality.org*” an open source library implementation using C#.NET will be available.

References

1. Crossley, M., Kings, N.J., Scott, J.R.: Profiles — Analysis and Behaviour. BT Technology J. 21, 56–66 (2003)
2. Newbould, R., Collingridge, R.: Profiling Technology. BT Technology J. 21, 44–55 (2003)
3. Zhiwen, Y., Xingshe, Z., Yanbin, H., Jianhua, G.: User Profile Merging Based on Total Distance Minimization. In: Proceedings of the 2nd International Conference On Smart homes and health Telematics (ICOST 2004), pp. 25–32. IOS Press, Singapore (2004)
4. Bruns, V., Reymann, S.: Development of a Middleware Solution for Consumer Profiling and Advanced Profile Distribution. Master of Science Thesis, Tampere University of Technology, Tampere (2007)
5. Namu Research Group, <http://namu.cs.tut.fi/>
6. Friend of a Friend Project, <http://www.foaf-project.org/>
7. Description of a Career, <http://ramonantonio.net/doac/>
8. Media RSS, <http://search.yahoo.com/mrss>
9. iTunes Library, <http://www.apple.com/DTDs/PropertyList-1.0.dtd>
10. Lehtikoinen, J., Aaltonen, A., Huuskonen, P., Salminen, I.: Personal Content Experience - Managing Digital Life in the Mobile Age. Wiley-Interscience, Chichester (2007)